[Re] Neural Network Model of Memory Retrieval

de la Torre Ortiz, Carlos

2021

de la Torre Ortiz , C & Nioche , A 2021 , ' [Re] Neural Network Model of Memory Retrieval ' , ReScience C , vol. 6 , no. 3 . https://doi.org/10.5281/zenodo.4461767

http://hdl.handle.net/10138/335176
https://doi.org/10.5281/zenodo.4461767

RESCIENCE C

Replication / Computational Neuroscience

[Re] Neural Network Model of Memory Retrieval

Carlos de la Torre-Ortiz[1,2] and Aurélien Nioche[2]

[1]University of Helsinki, Department of Computer Science, Helsinki, Finland – [2]Aalto University, Department of Communications and Networking, Helsinki, Finland

Edited by
Olivia Guest

Reviewed by
Sebastian Bobadilla-Suarez
James Roach

Received
08 November 2019

Published
25 January 2021

DOI
10.5281/zenodo.4461767

1 Introduction

Memory is the ability to store and retrieve information. We can distinguish procedural memory and declarative memory. Procedural memory is a type of memory that does not require conscious recall and is mostly related to motor tasks, while declarative memory is the ability of a conscious recall of information. Declarative memory can be itself divided into subcategories: semantic and episodic memory. Episodic memory stores past experiences and their emotional associations, while semantic memory stores and recalls facts independent of the context [1].

Memory, and especially semantic memory, can be tested in several ways. In an associative learning task, two stimuli are mapped together e.g. two words. The subject is then presented with one element of the stimuli pair and has to recall the corresponding other — that is, to recall from a partial cue. Another test to assess memory is with free recall tasks. In this type of task, a subject is presented with a set of items to memorize. Later, the subject is asked to recall as many items as possible [2].

Previous literature has shown that recalling memory items in the absence of cues is a difficult task: subjects usually fail to recall more than short lists of items in a free recall task [3]. However, according to the *Search of Associative Memory* (SAM) model, associations between memory items influence memory recall even in the absence of partial cues. From a neuroscience perspective, this could be explained by the overlaps between neuronal representations of memories [4, 5].

Recanatesi *et al.* [6] present a model of memory retrieval based on a Hopfield model for associative learning, with network dynamics that reflect associations between items due to semantic similarities.

Indeed, transitions occur due to the activation of populations of neurons encoding for a memory item. This sequential activation of neuronal ensembles forms stable states at different domain regions of a periodic function, which provides inhibition to the network. Network dynamics are also compatible with empirical observations about free recall previously described [6].

In the present work, we proceed to replicate the model as presented by Recanatesi *et al.* [6]. During our replication efforts, we discover several errors in parameters and collaborate with the original work authors to provide a successful replication and correct the original article.

Correspondence should be addressed to Carlos de la Torre-Ortiz (carlos.delatorreortiz@helsinki.fi)
The authors have declared that no competing interests exist.
Code is available at https://github.com/c-torre/replication-recanatesi-2015. – SWH swh:1:dir:0b541aeb4707ecedfbbcdd85adfd0100d748cc03.
Open peer review is available at https://github.com/ReScience/submissions/issues/10.

2 Background

Hopfield [7] proposed a model in which memory storage and retrieval emerge as properties of the collective behavior of its units, or neurons. This connectionist model is capable of recovering a previously presented pattern or patterns from partial cues, being able to complete the missing information.
Hopfield networks behave as fixed-point attractor networks as their internal state evolves towards a stable single state or fixed point. This is given by their energy function. These types of systems have been used as models of associative memory [8].
In the classic model as described by Hopfield [7], neurons are binary units: the activation state of each neuron can be either firing or not (on or off). The activity of each unit asynchronously changes in a discrete time scale.
As in other connectionist systems, the strength of connections between nodes is described by its weight matrix. Weights are only updated upon network initialization and depend on the patterns presented to all network units. The weight matrix takes the shape of a square, symmetric matrix in which all the values in the main diagonal are always zero. All neurons are connected to each other. Also, every neuron is both an input and output node for memory pattern presentation and retrieval. To compose the weight matrix, each node is updated according to a local incremental learning rule, related to Hebbian learning. Hebb's rule states that neurons that fire together when a certain pattern is present strengthen the connections between them [9].
In the work by Recantesi *et al.* [6], modifications to the original Hopfield model have been introduced, with new properties of memory retrieval: the model was adapted to induce transitions between attractor states (recalled memories).

2.1 Neuron Dynamics

Current — The original paper described the dynamics of neuron ν_i as the change of its current c_i with time:

$$\tau \dot{c}_i(t) = -c_i(t) + \sum_{j=1}^{N} r_j(t) \cdot W_{i,j} + \xi_i(t) \tag{1}$$

where:

$\tau \in \mathbb{R}$ is the decay time,
$c \in \mathbb{R}$ is the synaptic current,
$N \in \mathbb{N}$ is the network number of neurons,
W is the weight matrix,
$r \in \mathbb{R}$ is the firing rates,
$\xi \in \mathbb{R}$ is the Gaussian noise.
The former equation can be discretized using the Euler method:

$$c_i(t+1) = -c_i(t) + \frac{dt}{\tau}\left[-c_i(t) + \sum_{j=1}^{N} r_j(t) \cdot W_{i,j} + \frac{\xi_i(t)}{\sqrt{dt}}\right] \tag{2}$$

where:
$dt \in \mathbb{R}$ is the integration time step

Firing Rates — Firing rates r of each neuron are calculated by the gain function $g(c)$, a step function with sublinear behavior or a value of zero:

$$g(c) = \begin{cases} (c+\theta)^{\gamma} & \text{if } (c+\theta) > 0 \\ 0 & \text{if } (c+\theta) \leq 0 \end{cases} \tag{3}$$

where:

$\theta \in \mathbb{R}$ is the gain function threshold,
$\gamma \in \mathbb{R}_{<1}$ is the gain function exponent.

Memory Patterns — Every memory is represented by a binary vector or pattern $\boldsymbol{p}$ of size N. Each element of this vector corresponds to the state s of each neuron ν for that memory pattern. The value of this state is 0 if the neuron does not encode for that pattern and 1 if it does.
The different states are stored in a $M \times N$ matrix, with shape:

$$\begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_m \end{array} \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \cdots & \nu_n \end{array} \\ \begin{pmatrix} s_1^1 & s_2^1 & \cdots & s_n^1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^m & s_2^m & \cdots & s_n^m \end{pmatrix} \end{array}$$

Inhibition — The network is subjected to periodic inhibition driven by a sine wave $\phi(t)$, with the form:

$$\frac{\phi_{max} - \phi_{min}}{2}\left[1 + \sin\left(2\pi t + \frac{\pi}{2}\right)\right] \tag{4}$$

where:

$\phi_{min} \in \mathbb{R}$ is the minimum inhibition hyperparameter,
$\phi_{max} \in \mathbb{R}$ is the maximum inhibition hyperparameter,

Weights — Each neuron in the network is fully connected to all the other neurons. This gives a $N \times N$ weight matrix $\boldsymbol{W}_{i,j}$ representing the strength of connection or weight w between neurons ν_i and ν_j:

$$\begin{array}{c} \\ \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{array} \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \cdots & \nu_n \end{array} \\ \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix} \end{array}$$

To calculate the weight matrix, the following Hebbian rule is used:

$$W_{i,j} = \frac{\kappa}{N}\left[\sum_{p=1}^{M}(s_i^p - f)(s_j^p - f) - \phi(t)\right] \tag{5}$$

where:

$\kappa \in \mathbb{R}_{\geqslant 0}$ is the excitation,
$M \in \mathbb{N}$ is the number of memories,
$s \in \mathbb{Z}_{[0,1]}$ is the neuron state for a memory,
p is the memory pattern,
$f \in \mathbb{R}$ is the sparsity,
$\phi \in \mathbb{R}$ is the oscillatory inhibition.

To account for short term associations to the previous and next memories as in the SAM model, a new term $\boldsymbol{W}^*_{i,j}$ is added to the original weight matrix:

$$W^{SAM}_{i,j} = W_{i,j} + W^*_{i,j} = W_{i,j} + \frac{\kappa}{N}\left[\kappa_f \sum_{p=1}^{M-1} s_i^p \cdot s_j^{p+1} + \kappa_b \sum_{p=2}^{M} s_i^p \cdot s_j^{p-1}\right] \tag{6}$$

where:

$\kappa_f \in \mathbb{R}$ is the forward contiguity,
$\kappa_b \in \mathbb{R}$ is the backward contiguity.

Noise — Each neuron is subjected to Gaussian noise ξ, following the probability density function:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{7}$$

where:

$\mu \in \mathbb{R}$ is the noise mean,
$\sigma \in \mathbb{R}_{\geqslant 0}$ is the noise standard deviation.

2.2 Population Dynamics

Simulating the network with the original parameters is very computationally expensive, with the computation time depending primarily on the number of neurons. The system can be simplified, reducing the number of simulated units. All neurons that present the same activation state for the different memories will be considered that belong to the same population π. Moreover, all these neurons have an identical weight matrix $\boldsymbol{W}_{i,j}$[1], which will be the weight matrix of the population.

Current — A new term S_π is introduced in the calculation:

$$S_\pi = \frac{N_\pi}{N} \tag{8}$$

where:

$N_\pi \in \mathbb{N}$ is the number of neurons in a population.

The change of current of population π_i with time is:

$$\tau \dot{c}_i(t) = -c_i(t) + \sum_{j=1}^{U} r_j(t) \cdot W_{i,j} \cdot S_\pi + \xi_i(t) \tag{9}$$

where:

$U \in \mathbb{N}$ is the number of unique populations.

[1] ij notation refers to the simulated unit, which is from now on a population of neurons instead of the single neuron.

After discretizing by Euler:

$$c_i(t+1) = -c_i(t) + \frac{dt}{\tau}\left[-c_i(t) + \sum_{j=1}^{U} r_j(t) \cdot W_{i,j} \cdot S_\pi + \frac{\xi_i(t)}{\sqrt{dt}}\right] \tag{10}$$

Firing Rates — Firing rates are calculated with the gain function as before.

Memory Patterns — The $M \times N$ matrix containing s states is now a $M \times U$ matrix. This second matrix contains fewer elements than the original matrix, allowing for faster computation.

Inhibition — Inhibition calculation remains unchanged for the simulation with populations.

Weights — The population weight matrix $\boldsymbol{W}_{i,j}$ for neuron populations, with $U \times U$ shape:

$$W_{i,j} = \frac{\kappa}{N}\left[\sum_{p=1}^{M}(u_i^p - f)(u_j^p - f) - \phi(t)\right] \tag{11}$$

$$W_{i,j}^{SAM} = W_{i,j} + W_{i,j}^{*} = W_{i,j} + \frac{\kappa}{N}\left[\kappa_f \sum_{p=1}^{M-1} u_i^p \cdot u_j^{p+1} + \kappa_b \sum_{p=2}^{M} u_i^p \cdot u_j^{p-1}\right] \tag{12}$$

where:

$u \in \mathbb{Z}_{[0,1]}$ is the activity state of a population.

Noise — Unmodulated Gaussian noise ξ is computed as before.

2.3 Simulation

Simulation is carried away with population-level conditions (subsection 2.2). Calculations are then based on a $M \times U$ matrix instead of a much larger $M \times N$ matrix. Computation time now scales with M instead of N but for very large values of N, which would also increase the number of populations. Recanatesi *et al.* [6] estimated this method to be 99.9% faster than simulating individual neurons.
Firing rates are initialized at r_{ini} for populations encoding a randomly chosen memory pattern. Currents are set at $r_{ini}^{\frac{1}{\gamma}}$. All weights are also defined at this stage. Values of noise and inhibition change per time step. Neuron currents and firing rates are then calculated at each time step as well.
A memory is considered recalled if the average firing rate of all encoding neurons is above r_{recall}. The network is said to recall a certain memory p if the former condition is ever fulfilled for that memory during the simulation time.
Table 1 summarizes all hyperparameters used in the simulation.

2.4 Recall Analysis

The network was simulated first at a smaller scale and for one trial to observe detailed dynamics. It was then scaled to computer clusters, allowing to reach a total of 10,000 simulations for each condition needed for memory recall analysis. Each network is simulated for a total of 450 time cycles.
Several metrics are computed to assess the recall performance of the model. In particular, inter-retrieval time (IRT) is calculated as the number of time cycles until the recall of a new memory item. Other performance metrics such as memory size intersections or the average total recalls are also analyzed.

2.5 Computational Tools

Replication was carried out using Python 3.8.5 with packages NumPy 1.19.1, pandas 1.0.5, matplotlib 3.3.0, SciPy 1.5.2, and tqdm 4.48.0 on Parabola GNU/Linux-libre and CentOS GNU/Linux.

3 Results

3.1 Model Simulation

Current Dynamics During periods of minimum inhibition, a set of populations displays a positive current corresponding to one memory. Meanwhile, the remaining populations have negative current, meaning other memories are not being recalled.
At inhibition maxima, transitions between attractors may happen, with a new set of neuron populations firing at the next inhibition minimum. Transitions are observed when current values near inhibition minima, where values are close enough for the noise to drive changes between limit cycles (Figure 1).

Firing Rates Currents are subjected to a step function to calculate firing rates. The curve shape of the latter is then closely related to the positive domain of current values over time. Firing rates above r_{recall} indicate that a memory was recalled. Different memories are recalled at times corresponding to minimum values of inhibition. Transitions may happen between memory items or attractors in periods of minimum inhibition (Figure 2).

Inhibition The sine wave function provides the network with oscillatory inhibition necessary for its dynamics. Values have to be adequately scaled to induce the appropriate network behavior of memory recall and transitions between attractors (Figure 3).

Weights Weights show the strength of the connection between elements ij of the matrix. In the model, three different weight matrices are presented, accounting for the regular connectivity between neuron populations, but also considering item contiguity or associations between Weight values change according to the parameters of excitation, forward, and backward contiguity (Figure 4).

Noise Uncorrelated Gaussian noise is calculated for each population of neurons. The range of values is critical for the network to be successfully simulated, observing the transition between attractors (Figure 5).

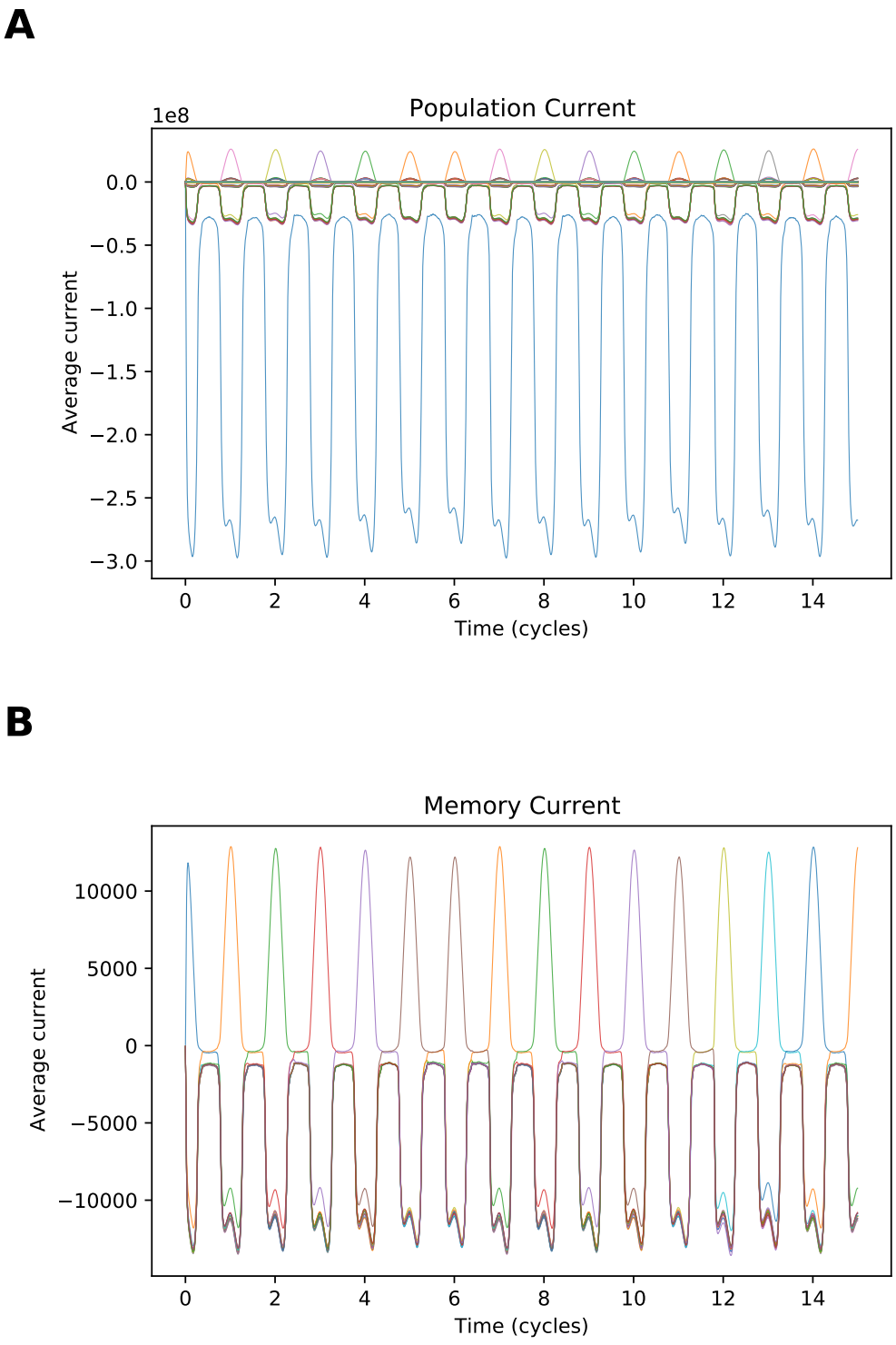

Figure 1. Currents. **A.** Currents of each population of neurons over time. **B.** Memories activation over time. Each color represents a different population (A) or memory (B). Axis units are arbitrary.

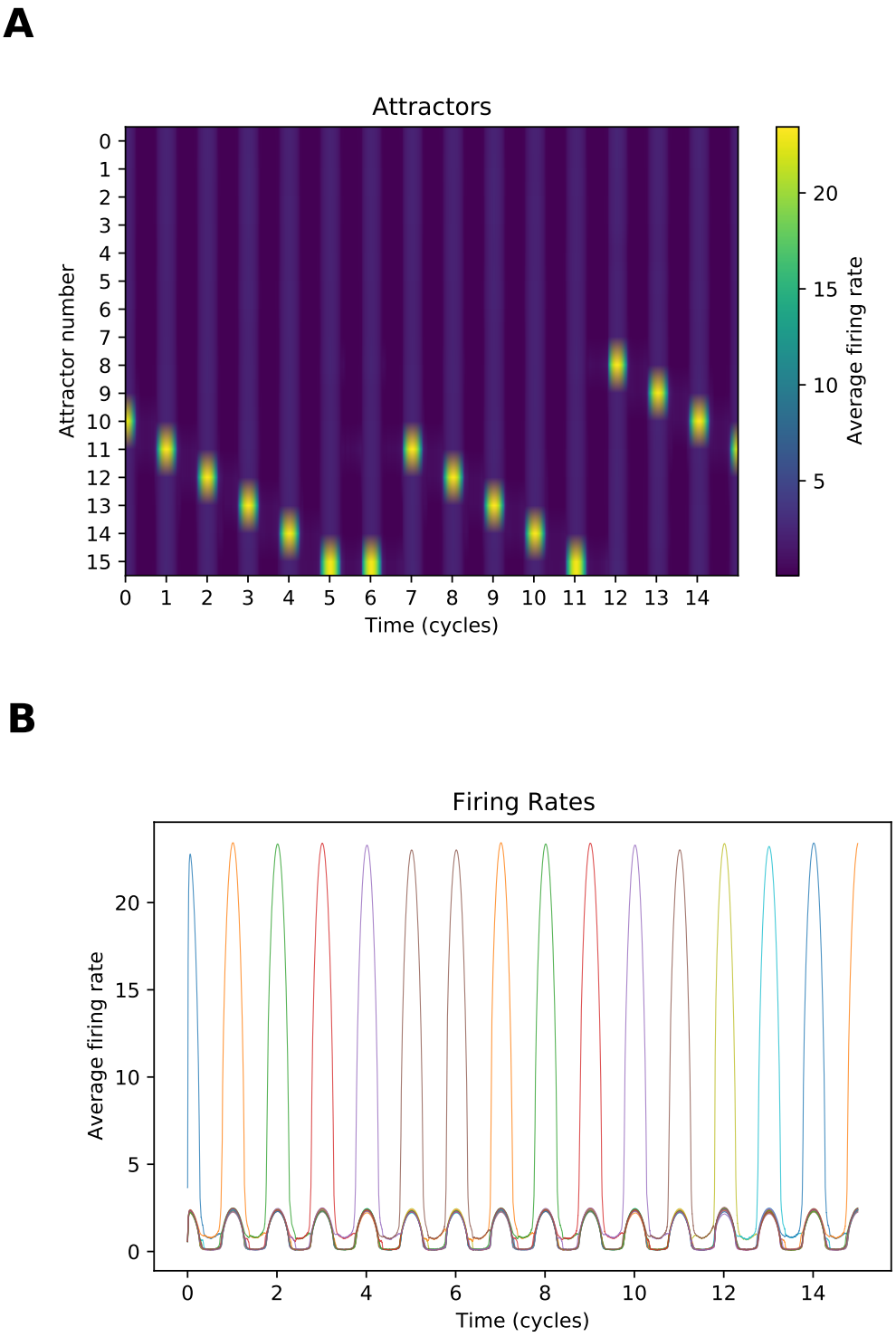

Figure 2. Network dynamics. **A.** Attractor states. The color indicates the firing rate. **B.** Average firing rates corresponding to each memory pattern. Each color represents a population. Transitions may happen between memory items or attractors in periods of minimum inhibition. Axis units are arbitrary.

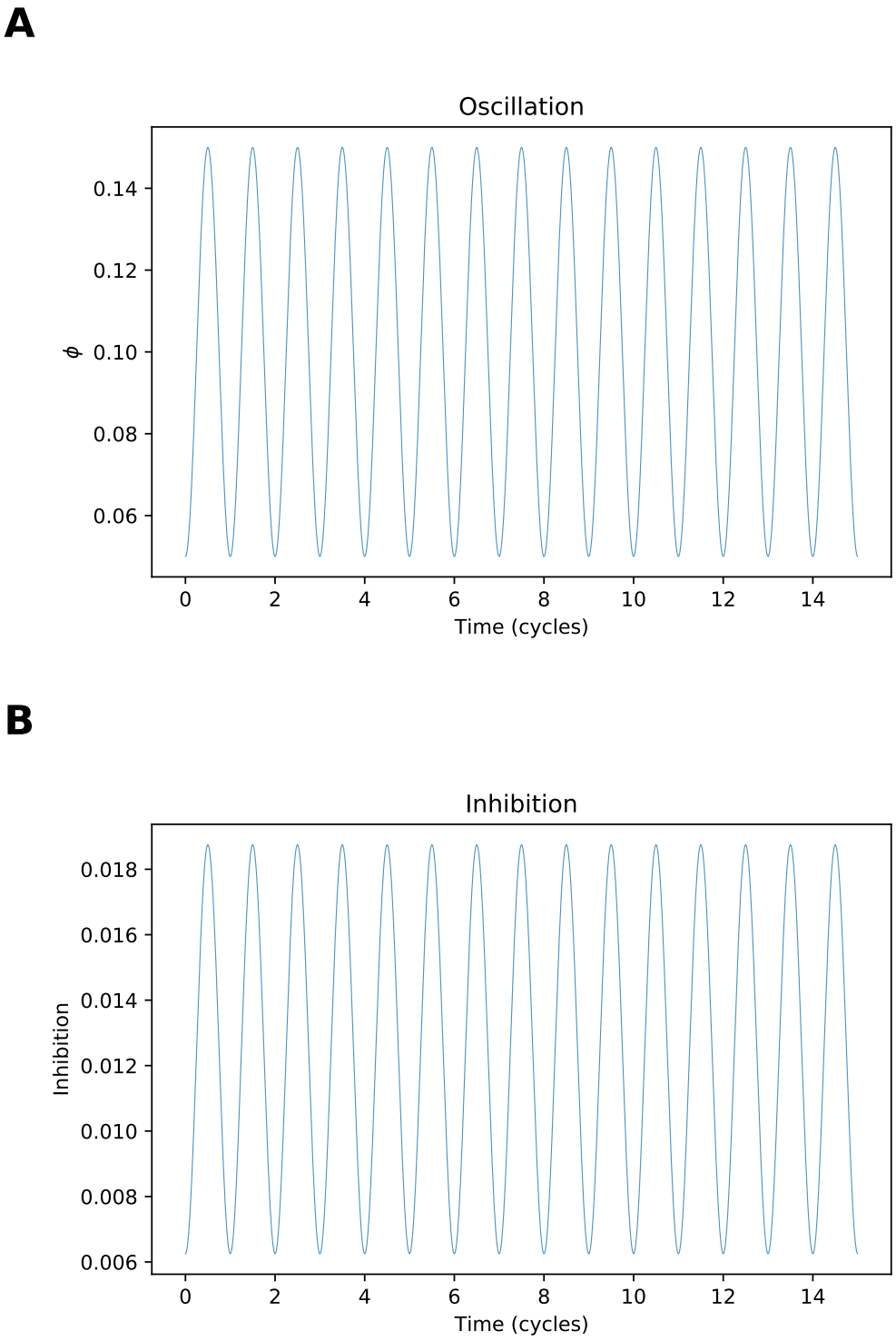

Figure 3. ϕ function and inhibition. **A.** Sine wave function values over time, which need to be scaled to have an adequate inhibitory effect on the network. **B.** Inhibition over time, driving the periodic behavior of the network. Axis units are arbitrary.

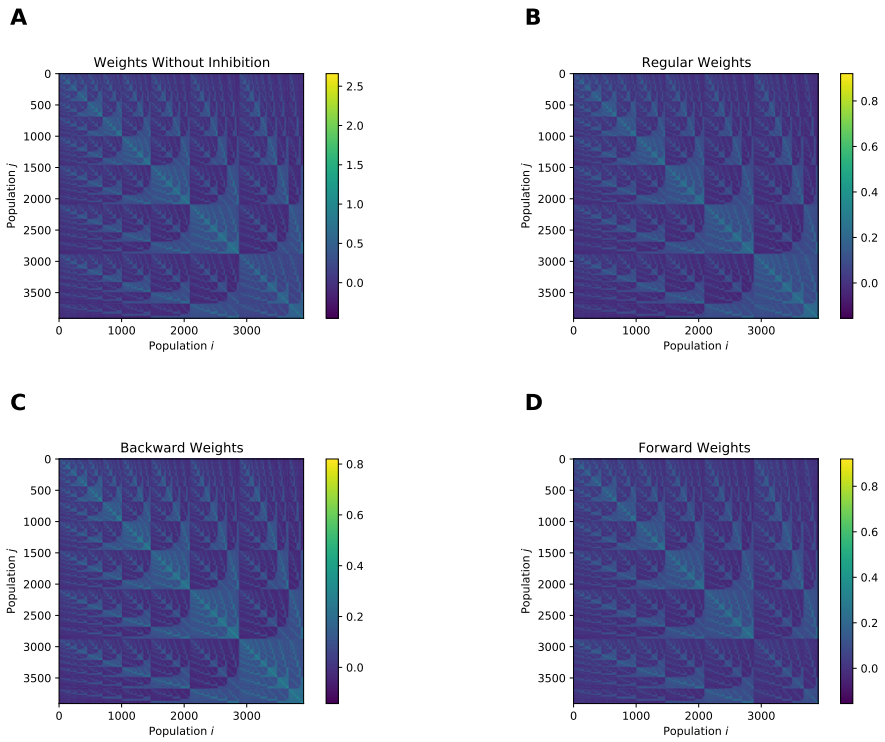

Figure 4. The strength of the connection between network elements is shown, with higher values in the color scale indicating stronger association (see color bars). **A.** Weight matrix previous to the addition of the inhibitory terms. **B.** Weights after adding inhibition. **C.** Weights corresponding to backward item contiguity. **D.** Weights corresponding to item contiguity. Values before applying inhibition (A) are higher than those in "regular" and contiguity connectivity (B, C, D). Also, the overall distribution of values is shifted with respect to the main diagonal in backward (forward) contiguity connectivity, as the connection links to the previous (next) unit. Axis units are arbitrary.

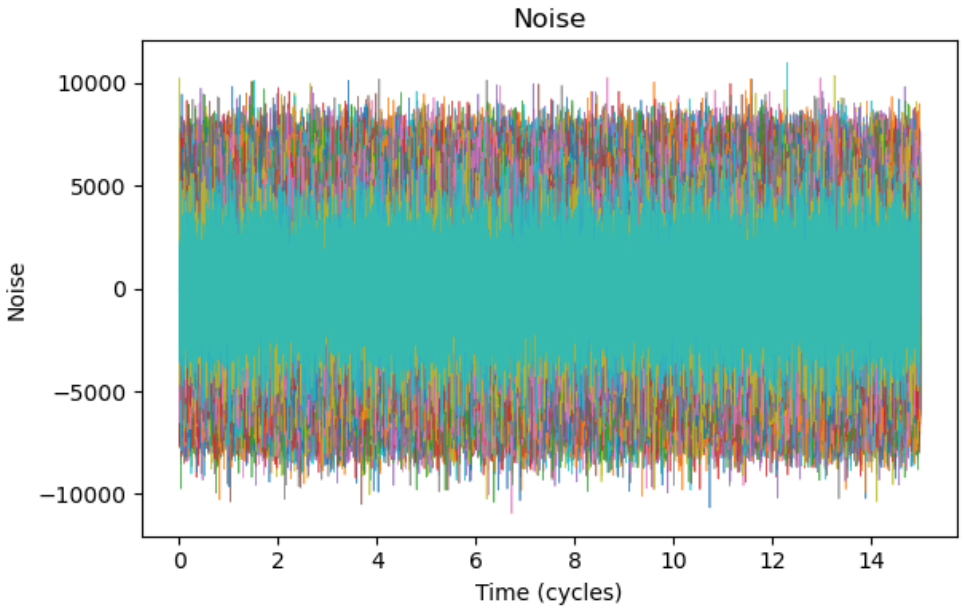

Figure 5. Noise for all populations, with a different color for each population of neurons. Values are centered around 0 (the mean of the distribution), and maximum and minimum values fall within a range that allows attractor transitions without distorting basic network dynamics. Axis units are arbitrary.

3.2 Recall Analysis

Temporal Properties of Recall Recall of new memories progressively slows down with time, while still possible even at later time cycles, even observing a sharp increase in recalls in the last iterations. Most transitions occur after one time step (IRT = 0), while there is some variability in the distribution. As time passes, the average IRT is likely to decrease due to these rapid memory transitions (Figure 6).

Probability of Recall The frequency of recall monotonically increases with memory size, as more overlaps between large memories are expected. As time passes, the average IRT is likely to decrease due to these rapid memory transitions (Figure 7). Differences in the number of points of the figure compared to the original article are likely due to binning and not due to a change in dynamics.

Memory Transitions More similar memories are expected to be recalled more often. This effect is observed as most transitions occur between the most similar memories. There is also a higher transition rate between memories with a lower intersection size between them. This leads to fast transitions, or lower IRT values for more similar memories (Figure 8).

Recall Performance and Parameters Average total number of memories recalled by 100 networks for 100 different values of forward contiguity and noise variance (Figure 9). Networks recall more words on average with monotonically increasing with the value of noise variance σ^2 until saturation. The performance also increases along with forward contiguity κ_f until saturation, followed by a decrease in recalls. At this point, the contiguity term likely overcomes noise as the drive for memory transitions. An additional evaluation focuses on lower forward contiguity values κ_f. This supports that the dynamics at the lowest values of κ_f in the previous figure are due to its relationship with κ_b, and not entirely due to randomness. A drop in performance is seen as values get closer to backward contiguity $\kappa_b = 850$, rising again afterward.

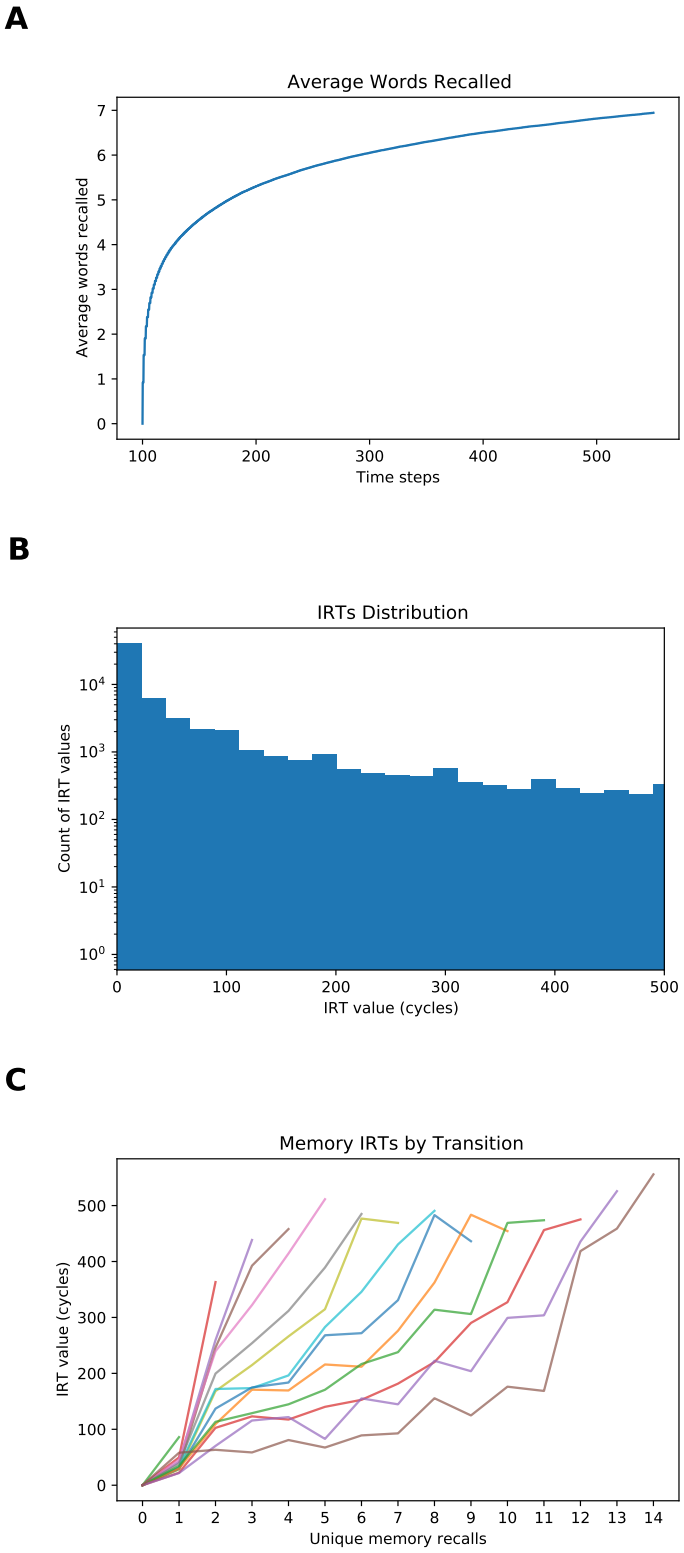

Figure 6. Temporal properties of recall. **A.** Cumulative sum of new memory recalls: a memory is added only the first time it is recalled over time. **B.** Number of occurrences of IRTs ordered by their size in time cycles. **C.** Average IRT values divided by the number of transitions for each line (memory).

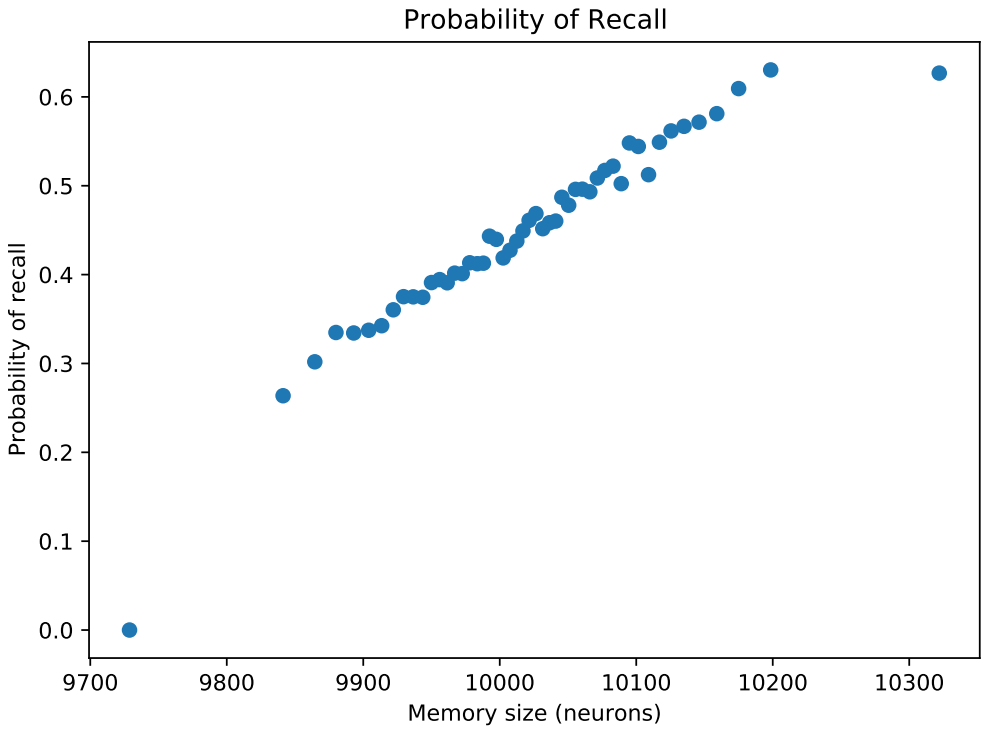

Figure 7. Frequency of memory recalls according to their size. Larger memories are recalled more often.

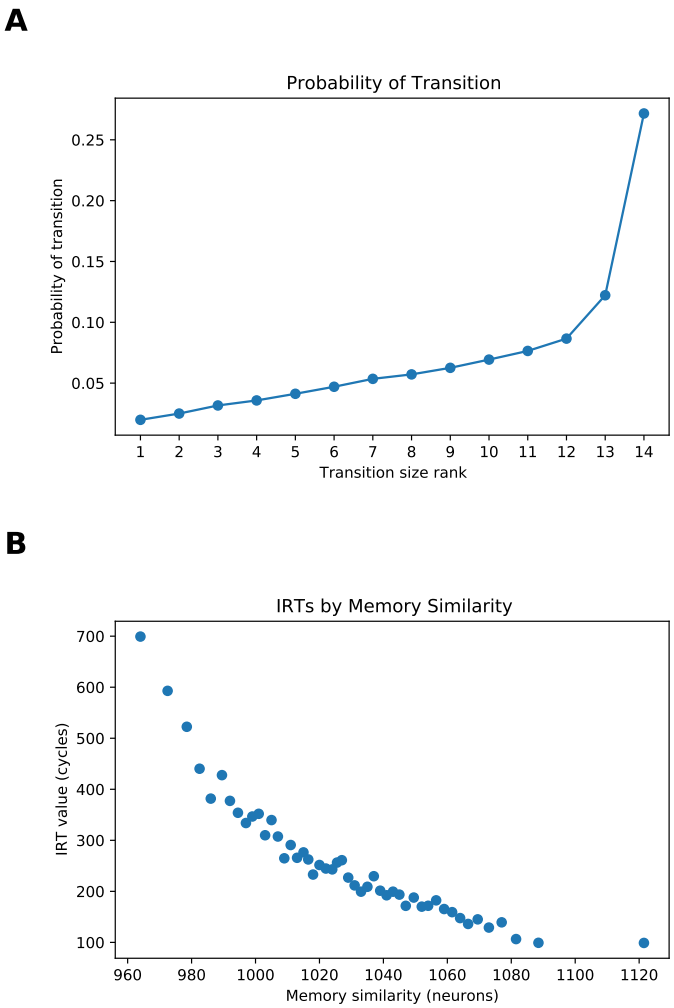

Figure 8. Influence of memory intersection sizes in the recall process. **A.** Proportion of transitions ranked in 15 groups of the same size, from less to more similar. **B.** Average IRT as a function of the intersection size in neurons

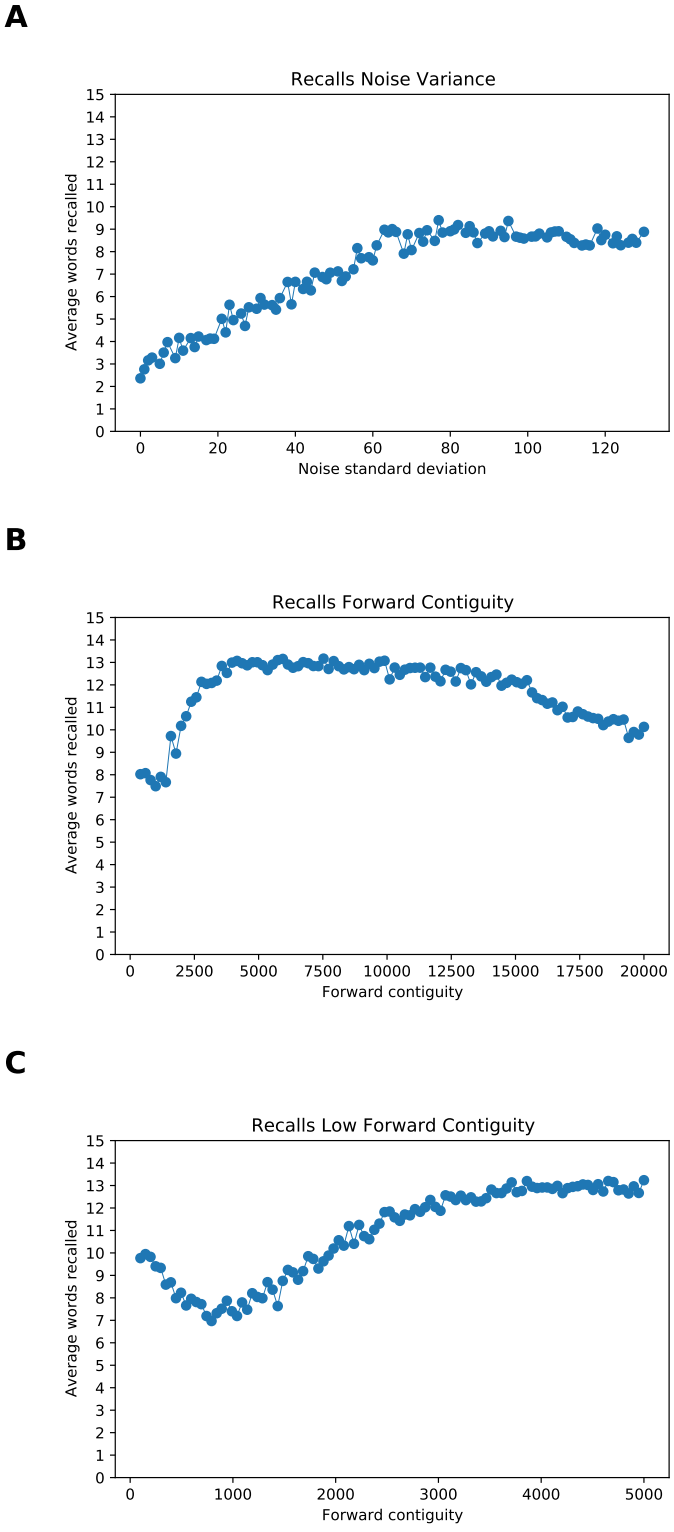

Figure 9. Recall performance with varying forward contiguity and noise variance. **A.** Performance when varying the value of noise variance σ^2 between 0 and 130. **B.** Performance when varying the value of forward contiguity κ_f between 850 and 20,000. **C.** Recall performance with varying forward contiguity κ_f between 100 and 5,000. A drop in performance is seen as values get closer to backward contiguity $\kappa_b = 850$, rising again afterward.

4 Discussion

Recanatesi *et al.* [6] present a neural network model of long-term memory free recall. In this model, inhibitory oscillations drive network dynamics. Noise and memory item contiguity can change the active attractor.
We were not able to replicate the model with the conditions of the original article. The original authors acknowledged several errors in their manuscript, making replication unlikely. Fortunately, collaboration with the original authors enabled to reach successful replication.
Most changes involve a normalization in equation terms, leading to changes of several orders of magnitude. An error in the original article scaling both contiguity parameters corresponding to equations 6 and 12 in the reference paper. In the corrected version, a previously missing pre-factor provides correct normalization as reported in equations 6 and 12.
Besides, the base values of several hyperparameters needed to be corrected as reflected in Table 1. The following parameters were changed: γ, κ, κ_f, and κ_b. Parameters of the original article allowed to replicate of retrieval dynamics, but could not replicate the recall analysis.
After applying the corrections in coordination with the original authors, we do not observe important differences with the original article, reporting a full replication of the original results.

5 Conclusions

In this work, we successfully reproduced the results of the memory model simulation reported by [6]. This was possible by modifying the equations of the original article. Changes scale parameters or modify their base values to correct the errors in the original manuscript in coordination with the original authors. As in the reference research, we show that oscillating inhibition, together with noise and item contiguity, induce the transition of recall of different memories in a Hopfield model of memory retrieval.

Acknowledgments

We acknowledge the computational resources provided by the Aalto Science-IT project and University of Helsinki Finnish Grid and Cloud Infrastructure (persistent identifier urn:nbn:fi:research-infras-2016072533).

Parameter	Description	Value
N	Number of neurons	100,000
M	Number of memory patterns	16
τ	Decay time	0.01
θ	Gain function threshold	0
γ	Gain function exponent	$1/3$
f	Sparsity	0.10
κ	Excitation parameter	12,500
κ_f	Forward contiguity	1,500
κ_b	Backward contiguity	850
ϕ_{min}	Minimum inhibition parameter	0.4
ϕ_{max}	Maximum inhibition parameter	1.2
μ	Noise mean	0
σ	Noise standard deviation	$\sqrt{65}$
t_{cont}	Simulation time, continuous	450
dt	Integration time step	0.001
r_{ini}	Initial encoding rate	1
r_{recall}	Recall activation threshold	15

Table 1. Hyperparameters and reference values

References

1. L. R. Squire. "Memory and Brain Systems: 1969–2009." In: **J Neurosci** 29.41 (2009), pp. 12711–12716. DOI: 10.1523/JNEUROSCI.3575-09.2009.
2. E. Tulving and F. I. M. Craik. **The Oxford Handbook of Memory**. Oxford: Oxford University Press, 2000.
3. B. B. M. Jr. "The immediate retention of unrelated words." In: **J Exp Psychol** 60 (1960), pp. 222–234. DOI: 10.1523/JNEUROSCI.3575-09.2009.
4. J. G. W. Raaijmakers and R. M. Shiffrin. "SAM: A Theory of Probabilistic Search of Associative Memory." In: **Psychol. Learn. Motiv.** 14 (1981), pp. 207–262. DOI: 10.1016/S0079-7421(08)60162-0.
5. S. Romani, I. Pinkoviezky, A. Rubin, and M. Tsodyks. "Scaling Laws of Associative Memory Retrieval." In: **Neural Computation** 25.10 (2013), pp. 2523–2544. DOI: 10.1162/NECO_a_00499.
6. S. Recanatesi, M. Katkov, S. Romani, and M. Tsodyks. "Neural Network Model of Memory Retrieval." In: **Frontiers in Computational Neuroscience** 9 (2015), p. 149. DOI: 10.3389/fncom.2015.00149.
7. J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: **Proc Natl Acad Sci** 79.8 (1982), pp. 2554–8. DOI: 10.1073/pnas.79.8.2554.
8. D. J. Amit. **Modelling Brain Function: The World of Attractor Neural Networks**. Cambridge: Cambridge Univ. Press, 1992.
9. D. O. Hebb. **The Organization of Behavior; A Neuropsychological Theory**. New York: Wiley, 1949.